

Aberystwyth University

Adaptive and survivable trust management for Internet of Things systems

Jabeen, Farhana; Khan, Zia-ur-Rehman; Hamid, Zara; Rehman, Zobia; Khan, Abid

Published in:
IET Information Security

DOI:
[10.1049/ise2.12029](https://doi.org/10.1049/ise2.12029)

Publication date:
2021

Citation for published version (APA):

Jabeen, F., Khan, ZR., Hamid, Z., Rehman, Z., & Khan, A. (2021). Adaptive and survivable trust management for Internet of Things systems. *IET Information Security*, 15(5), 375-394. <https://doi.org/10.1049/ise2.12029>

Document License CC BY

General rights

Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

ORIGINAL RESEARCH PAPER

Adaptive and survivable trust management for Internet of Things systems

Farhana Jabeen¹  | Zia-ur-Rehman Khan¹ | Zara Hamid¹ | Zobia Rehman¹ | Abid Khan²

¹Computer Science Department, COMSATS University, Islamabad, Pakistan

²Computer Science Department, Aberystwyth University, UK

Correspondence

Farhana Jabeen, Computer Science Department, COMSATS University, Islamabad, Islamabad Capital Territory 45550, Pakistan.
Email: farhanakhan@comsats.edu.pk

Abstract

The Internet of Things (IoT) is characterized by the seamless integration of heterogeneous devices into information networks to enable collaborative environments, specifically those concerning the collection of data and exchange of information and services. Security and trustworthiness are among the critical requirements for the effective deployment of IoT systems. However, trust management in IoT is extremely challenging due to its open environment, where the quality of information is often unknown because entities may misbehave. A hybrid context-aware trust and reputation management protocol is presented for fog-based IoT that addresses adaptivity, survivability, and scalability requirements. Through simulation, the effectiveness of the proposed protocol is demonstrated.

1 | INTRODUCTION

The Internet of Things (IoT) is a technological revolution in the world of computing and communication, enabling advanced services where (physical and virtual) things in multiple domains (such as industrial, health, commerce, and home) are increasingly being renewed from isolated systems to networked Internet-enabled devices. The vision is to connect heterogeneous things with varying interests and capabilities together, thus empowering them to collaborate and allowing people to communicate with them [1]. Reasons for collaboration include the following: (i) things might not be self-sufficient and may need services provided by other things to accomplish their goals/tasks, (ii) things might have limited knowledge regarding their surrounding environment, and (iii) things (entities) might have low capabilities in terms of computation, communication, and memory resources. Examples of collaboration scenarios include event monitoring using distributed query processing in wireless sensor network (WSN) [2, 3], reliable end-to-end packet delivery [4], IoT-based healthcare solutions [5], production supply chain [4], environmental monitoring [3], and agriculture [6].

IoT refers to dynamic and ever-evolving open environments where entities do not necessarily know each other. Therefore, decision-making related to selecting an entity for

collaboration is associated with uncertainty and risk. In case of uncertainty, an entity has a choice between two or more alternatives, cannot confidently predict the consequences of selecting any specific entity. Let us consider example scenarios. For example, the service consumer entity (truster) needs to know which service provider entity (trustee) in the network to trust for reliable packet forwarding for the routing process. For checking anomalous measurements or distributed in-network maximum temperature computation of a region, the truster needs to know which neighbouring entities to trust [7–11].

Due to the open environment of IoT, collaboration may open the door to internal and external attacks that affect the performance/accuracy of the entire system. Therefore, collaborations should be performed on a controlled basis. In this digital era, trust can be used as a measure to deal with uncertainty problems. Grandison et al. [12] defined trust as ‘*the firm belief in the competence of an entity to act dependably, securely and reliably within a specified context*’.

Hard trust relationships are based on cryptographic mechanisms, which allow external security attacks to be addressed [13]. The hard trust protects entities from vulnerabilities and attacks by, among others, allowing access only to authorized entities. Entities in IoT may be resource constrained and therefore unable to tolerate the load of heavy computations of cryptography-based protocols. However, false or

inaccurate information may be provided by authorized internal entities. Hard trust mechanisms do not continuously monitor participating entity behaviour. To ensure trustworthy cooperation between entities, soft security employs social control aspects to the underlying security mechanism [7, 9–11]. For example, soft trust can be based on direct experience (direct trust), trustworthy peer experiences collected during the period (indirect trust), or both. A trust engine called the *Trust and Reputation Management* (TRM) system explicitly computes the level of trust.

The TRM system gathers information from the entities that consumed the service regarding the quality of service (QoS) received and computes the reputations of the entities that provided the service [7, 8, 14, 15]. In a distributed TRM system, the service consumer can learn the service provider's past behaviour through direct interactions. But if the service provider does not have sufficient direct interactions with the specific service consumer, it must rely on an indirect experience computed from the recommendations obtained from peers who have already directly interacted with the service provider [8]. Reputation can be computed based on direct trust, indirect trust, or both.

Reputation is an assessment based on the trusted party behaviour during direct past interactions with the trusting party or/and as reported by peers through recommendations or third-party verification. [16]. Reputations of the entities help to calculate how much these members can be trusted for a particular service. The behaviour of an entity as a trustee or recommender may change over time; relying on its past behaviour only may be dangerous. Therefore, reputations are continuously recalculated with time. In IoT, entities may use the TRM scheme for a variety of purposes, such as finding a reliable walking path [17], accessing e-health records [6], and exchanging services [8–11].

The TRM system must be prepared to cope with variable conditions and malicious entities. One of the important requirements of the TRM system is adaptivity support. Using TRM to address the adaptivity requirement allow trusters to determine changes in service providers or recommender behaviour. In IoT, entities are heterogeneous and part of multiple self-interest communities, and the expectation of reliable reports from them is therefore challenging. To negatively affect the reputation of the good service provider (bad-mouthing) or artificially improve the reputation of a bad service provider (ballot-stuffing), recommenders might provide false recommendations. The goal of malicious peers is to cause harm to specific members of the network or the entire system (e.g. they may disseminate virus-infected audio files). The trust framework must allow the entities to update their belief sets concerning environmental changes. For example, a service consumer might form confidence regarding a particular service provider by trusting it as an optimal choice for a particular service. Service consumer should be able to determine the change in service provider behaviour if it is modified. The reasons for changes in behaviour might be malicious but alternatively could result from reducing the number of resources (such as battery power). The presented scheme allows

the truster to select the service provider based on its current behaviour or a combination of both past and present. The recommenders are required to provide recommendations for a service provider based on past and current behaviour.

TRM system success depends on its efficiency in accurately evaluating the trustworthiness of entities as service providers and as recommenders [7, 8]. The trustworthiness of the entity as a recommendation source (termed credibility) is an essential parameter in a trust model. When a malicious recommender node provides an unfair recommendation, its credibility should decrease. Such reductions in credibility would allow survivability to be achieved—that is, malicious users would not be allowed to significantly impact normal system operations or the reputation of the service provider. Most existing works on social IoT (SIoT) [18–21] assume that friends are cooperative but do not consider the credibility of the friends while receiving recommendations.

Moreover, some works related to TRM in IoT [19–21] consider recommender reputation (trustworthiness as a recommender of a service provider) as a weightage to the recommender's service provider recommendation value. Those authors did not consider that an entity with a good reputation (trustworthiness) as a service provider is not necessarily a trustworthy recommender. Furthermore, recommender credibility in their models was not reduced in response to malicious behaviour by the recommender. In contrast, our work computes an entity's recommender credibility by considering the similarity of the truster rating of recommender behaviour with that provided for the recommender's peers who provided recommendations for the same service provider in a similar context in the same period. The credibility of the malicious recommender is decreased with each malicious attempt, which allows service provider trustworthiness to converge to the ground truth in the presence of malicious nodes.

The problem with TRM systems based on purely distributed infrastructures [12, 15] is that each entity keeps the information about a small portion of the entire network [12, 14]. Therefore the information regarding the number of potential service providers is limited to local knowledge [9, 20]. Because of the limited local knowledge, the probability of obtaining service from highly trustworthy service providers available in the network is low. Searching for a highly trustworthy service provider by flooding the network is expensive because service provider behaviour may change over time. TRM systems based on a centralized trustworthiness evaluation [10, 11, 22] are confronted with scalability and energy efficiency limitations. Transmitting raw feedback from every entity (based on a single interaction) to some destination (centralized server or cloud) external to the network for storage and trustworthiness evaluation may be prohibitively expensive and sometimes not possible given typical data collection rates, network types, and network sizes [10, 23].

Compared with cloud computing, fog computing provides additional capabilities in terms of delay, mobility, scalability, heterogeneity, and privacy [22, 24]. Fog computing supports edge computing [25]. Fog-based IoT allows the division of the network into subnetworks, each governed by a fog-node. An

IoT entity can then register itself with one of the fog servers. The fog server can be held responsible for aggregating the information from IoT entities and forwarding it to the cloud. In IoT, multiple entities might have varying interests related to the selection of service providers for a specific service, which requires deciding based on either local knowledge only, global knowledge, or both. For example, for a low-priority service (such as obtaining the current temperature reading), the node may opt to obtain the service from a provider whose trustworthiness is computed using local knowledge (i.e. by computing direct and indirect trust using a distributed approach). However, for a high-priority service, the node may prefer to obtain the service from a highly reputable network service provider that can be found using global knowledge (centralized approach exploiting cloud or fog).

This paper proposes a hybrid trust management framework that allows for trustworthiness evaluation between entities using centralized and distributed approaches. The nodes perform in-network processing to address the scalability requirement in the centralized approach and send only the fine-grained values to the fog-node for evaluation. For global reputation calculation of IoT devices, the fog-nodes forward their computed reputation scores to the cloud. Every node in the network is made autonomous to select the evaluation approach based on its requirements, network conditions, and environment. For example, to obtain services from a very reputable service provider, the node can contact the fog-node in its domain (subnetwork) for a list of potential providers with high reputations.

To summarize, the contributions of our work include the following:

1. A trust-based attack-resistant trust model for IoT is proposed. It enables an entity's trustworthiness as a service provider and credibility as a recommender to be computed. We have considered the following five trust-related attacks that can disrupt the TRM system: self-promoting, bad-mouthing, ballot-stuffing, opportunistic service, and discriminatory.
2. The proposed TRM protocol addresses the survivability requirement by incorporating regulations with credible sanction options and statistical techniques. The TRM protocol supports the convergence of the service provider's trustworthiness to the ground truth in the presence of malicious entities.
3. The TRM protocol addresses the scalability requirement by incorporating temporal in-network data aggregation scheme to reduce the overall amount of data uploaded to the fog servers by IoT entities.
4. Effectiveness of our TRM protocol is demonstrated using application scenarios related to service composition. A comparative analysis of our proposed TRM protocol is carried out against the trust protocols presented in [9–11, 22].

The remainder of the paper is organized as follows. Section 2 discusses the related work, Section 3 presents the system

model, and Section 4 presents the threat model. Section 5 presents the trust management protocol followed by the experimental evaluation in Section 6. Section 7 discusses future work and concludes the paper.

2 | RELATED WORK

In this section, we review recently proposed trust management protocols for IoT systems.

2.1 | Trust and reputation management in the Internet of Things

Saied et al. [10] proposed a centralized context-aware trust management system to manage cooperation among nodes. The context similarity is computed based on the contextual distance (each service is assigned a value). A node must submit feedback/rating for every service it has received, which may lead to substantial traffic throughout the entire network. The work does not consider the adaptive behaviour of the service provider while computing credibility. The peers who rated a service provider as good in the past because of the provision of good QoS are penalized because of the service provider's current bad behaviour. The history of recommendation quality scores is maintained, which results in substantial storage requirements. In contrast, our work considered recommender j rating behaviour in computing recommender j credibility. Najib et al. [17] review the existing literature on trust calculation methods in IoT.

Shayesteh et al. [23] proposed a centralized context-aware trust management scheme for computing entity trust to manage collaboration among entities. Dempster–Shafer theory of evidence is used to aggregate recommendations from entities. A node must submit every feedback to the cloud, which may lead to substantial traffic throughout the entire network.

The work presented in [26] identifies dishonest nodes and revokes their credentials in the vehicular ad hoc network scenario. The proposed trust model focuses on evaluating the trustworthiness or credibility of received messages related to event occurrences. Based on indirect experience, event trustworthiness is computed by considering the information attributes such as Location closeness, Data Integrity, Authentication, and Time closeness.

Mendoza et al. [27] proposed a distributed trust management scheme for IoT to mitigate on-off attacks (alternatively behaving as a good node and as a bad node). The node is punished with more than the reward for not providing a service. For each service, the rating assigned is based on the importance of the service in terms of the amount of processing and energy requirements.

Gu et al. [28] proposed a trust management architecture divided into three layers: sensor, core, and application. For specific purposes, trust management is performed at each layer. The sensor layer collects information from the physical world. The core layer connects the sensor layer to the application

layer. The application layer is used to process and store information. The parameters considered for trust evaluation include (i) trust on the sensor layer, (ii) trust on the core layer, and (iii) trust on the application layer. Formal semantics and fuzzy set theory are used for the realization of the trust mechanism.

Hussain et al. [22] presented for fog-based IoT context-aware feedback and feedback crawler system. Experimental evidence related to addressing reliability, adaptive behaviour, and survivability requirements is not provided.

J. Yuan et al. [24] proposed a feedback information fusion algorithm to assess edge devices based on objective information entropy theory [25]. The presented scheme does not compute the reputation of IoT devices. The trust components considered include (i) interactions with devices and (ii) the service quality provided by the edge device. The trust evaluation is performed on edge devices. The agent is responsible for aggregating the direct trust and the trustee device's feedback trust. In this approach, a scalability issue arises due to increased bandwidth requirements. Moreover, in the case of resource-constrained IoT, this approach may result in the network longevity issue. Furthermore, in this work, no experimental evidence is provided to address adaptivity and survivability requirements.

Truong et al. [29] presented a feedback-based trust evaluation model. Like [30], the work focuses on using knowledge, recommendation, and reputation as trust indicators. The authors described the metrics to be used to derive these three trust indicators. The work suggested that service providers be selected based on a community of interest and cooperativeness. How the three trust indicators will be aggregated is not well explained. Furthermore, in this work, no experimental evidence is provided.

Kowshalya et al. [31] presented a distributed trust model. The work assumes that the node having higher residual energy in a group of peers should be considered malicious (that carry out ON/OFF selective forwarding attack). Moreover, the scheme's efficacy related to the resilience of the proposed trust model against trust-related attacks is not presented.

In [32], a trust management scheme is presented for efficient service composition in IoT environments without considering social relationships. Direct trust between service consumer and recommender is considered as a recommender's credibility. A rating similarity (RS) measure is used to calculate the recommender credibility for those cases with no direct trust between the recommender and service consumer. The scheme requires each feedback to be sent to edge nodes for trust calculation. A service provider's trustworthiness score is updated after every interaction. The presented scheme requires reduced recommender credibility within some threshold to achieve service provider survivability. If a recommender provides a recommendation score lower than the computed trustworthy score, its credibility is decreased. Otherwise, if the recommender provides a higher recommendation score than the computed trustworthy score, its credibility increases.

2.2 | Trust and reputation management in the social Internet of Things

To fully achieve an effective social network of intelligent objects called the SIoT, there exists work that addresses fundamental aspects [30, 33, 34]. In [35], the authors reviewed key components of SIoT including architecture, and trust management. This work also provided a comprehensive overview of the SIoT environment and related challenges.

Xiao et al. [18] presented a trust model based on guarantor and reputation for SIoT environments. The centralized server stores the ratings from the entities (objects) and computes the reputations of the objects. Upon the provision of good-quality service, the service provider is assigned some credits. If the service provider acts maliciously, then it must give some credits. The reputation computed for each object by the server is stored in the object and can later be updated using the agents (that request the reputation server).

Bao et al. [19] proposed a dynamic trust management system for IoT considering social relationships. Objects or nodes have specific owners. The relationships considered among objects and owners include (i) Friendship, (ii) community, and (iii) ownerships. Trust is calculated based on honesty, cooperativeness, and community of interest. In this work, the proposed protocol's resilience to attacks on the TRM system has not been proven by evaluations. The proposed TRM system for the specific IoT environment considers only WSNs.

Bao et al. [9] proposed a trust protocol for IoT that improves on [19]. Trust evaluation is based on honesty, cooperation, and a community of interest. Network nodes are divided into diverse communities based on interest. The work assumes that nodes belonging to a community will have the same social interests, which may not always be true. A new storage management strategy is proposed to meet the scalability requirement.

Chen et al. [20] proposed a trust management system for service composition in a service-oriented architecture-based IoT system. The work only considers the social relationships between their owners and does not consider social relationships between objects. All the devices owned by the same person are assigned similar trust values. Moreover, the characteristics of the different devices must affect the trust value. Users/owners compute their own trust values for devices. The work assumes the availability of a high-end device for every user, which cannot be guaranteed in every network.

Chen et al. [21] presented an adaptive trust management protocol. The protocol presented in the work identifies the best protocol settings considering two design parameters, α and β . Parameter α represents the trade-off between recent direct trust and past direct trust, whereas parameter β represents the trade-off between recent indirect trust and past information. A look-up table is maintained and populated at a static time, listing the best settings for the two parameters over a range of input parameter values. Chen et al. [36] presented a trust management scheme for efficient service composition in SIoT environments. The scheme considered both QoS

(reputation and available energy) and social relationship factors [21] for trust composition.

Most of the work [9, 18–21] done in this area considers social relationships as a measure of the trustworthiness of a node as a recommender. The work assumes that friends are cooperative, and the cooperativeness value is computed as the ratio of the number of common friends and the total number of friends. Moreover, the trustworthiness between objects is measured by exploiting the social relationship between the object owners in a transaction. However, objects can build their own social networks that may differ from and be independent of their owners' social networks. None of the existing works consider the credibility of friends when receiving recommendations.

Nitti et al. [11, 37] presented two approaches related to trustworthiness management in the SIoT: subjective and objective. In the subjective trust model, each node computes the trust of its friend based on its own experience and recommendations from friends. In the subjective approach, each node manages the feedback from other nodes and stores such information for trust calculation. In the objective trust model, feedback information regarding every node is distributed and stored in a dynamic hash table structure. All nodes can view this information, but only a few special nodes known as pre-trusted objects manage this information.

In [11], the credibility of a recommender node is calculated based on parameters including the trustworthiness of a service provider, computational capability, relationship factor and number of transactions. In [37], a recommender node's credibility is calculated based on parameters including the centrality and direct trust of the recommender as well as the number of transactions. In the objective approach, a node must submit feedback for every service it has received. The scheme presented in [11, 37] for computing credibility is not impacted by his/her malicious behaviour as a recommender. Therefore, these schemes cannot achieve survivability.

Tormo et al. [38] designed a prototype system that dynamically selects a suitable reputation computation engine based on system conditions (such as the number of users, available bandwidth, available storage capacity, and number of feedbacks received) and required performance metrics.

In IoT, service provider reputation is computed based on providing lightweight service. The computation cannot be considered the same for high-weight service that requires more resources and increased availability. Reputation context can enhance systems by providing better granularity. Existing works [10, 30] consider the context awareness of the trust model in computing the trustworthiness of the service provider based on the type of service provided. Like the work presented in [10, 30]. The emphasis is on the computation of the service provider's reputation based on the type of service offered. Truong et al. [30] presented a trust model that considers three metrics: reputation, recommendation, and knowledge. The knowledge trust metric is divided into two subontologies, human-to-human and human-to-device. Ontology is used to represent user knowledge that may not be appropriate for limited-resource objects.

Abderrahim et al. [39, 40] presented context-based centralized trust management system for SIoT. In [40], context represents the type of service. For trust evaluation and QoS, three types of relationships are considered: system, social, and community based. Furthermore, the efficacy of the TRM system in terms of dynamic node behaviour and addressing attacks is not shown. Abderrahim et al. [39] assumed that each social community would be distinguished from others based on social interests and location. Each community will have an admin as the central entity and is responsible for computing and storing trust values of objects sharing the same community. The work uses Dirichlet distribution to compute trust at the admin level. A prediction model based on the Kalman filter is used to prevent on–off attacks. In this work, a node is considered malicious if it leaves the community without admin permission.

Gai et al. [41] presented a trust management system for the Social Internet of Vehicles (SIoVs). To address trust information tampering stored locally, the trustor and the service provider, with the signature of central trusted authority, store the trust-related information (cookie) locally. The direct trust is computed using rater cookie, centrality, and relationship information. To compute the indirect trust, firstly, direct trust towards service the trust-related information (cookie) is stored locally provider and each recommender mentioned in cookies (provided by service provider) is computed by trustor. Later, indirect trust is computed as the ratio of the summation of direct trust between service providers and recommenders to the summation of direct trust between service consumers and recommenders.

Jayasinghe et al. [42] presented a flow-based trust management system using the page rank [43] model. The work assumed that if two entities are linked, they have a relationship regardless of trust. This assumption is used to generate a weighted directed graph where edges represent the relationships among objects and vertices represent objects. In this work, if the number of outgoing links exceeds a certain threshold, the entity is considered a trustworthy entity. The recommendation value of an entity is calculated as the sum of the number of incoming links.

In [44], the authors focus on key challenges in designing trust management for SIoVs. Existing related literature is also reviewed. Moreover, the work provides a vision of using new technologies to design efficient SIoV trust models.

In [45] design of the trust management platform is presented for the SIoT environment. The authors suggested components for trust information collection and management, including analytics. The authors also proposed a generic reference model for IoT, which is comprised of four planes: (i) IoT trust and security plane, (ii) physical IoT plane, (iii) social/cyber IoT plane, and (iv) IoT management plane.

3 | SYSTEM MODEL

Like approaches defined in [17, 22, 24, 27, 31], we consider an IoT system where an entity can be resource constrained (equipped with limited storage) and, therefore, cannot store the

complete set of trust values towards other entities and the services they offer over the long period. A resource-constrained entity in IoT can maintain information (reputation, services, credibility) for the limited neighbours that lie in its allowed neighbourhood. For better management, scalability, and efficiency, the network is assumed to be divided into many subnetworks. Each subnetwork contains a resource-full fog-node (referred to as a kingpin node) and a set of entities, as shown in the system model presented in Figure 1. The entities form a connected graph with inter-connect topologies between them. Wired, wireless, and optical networking facilities can be exploited to construct such topologies. A kingpin node is installed at the close location that covers a specific spatial region and processes data gathered close to the network's edge. IoT entities located in a specific spatial region register with the kingpin node responsible for that region. All kingpin nodes work as subnetwork controllers (shown in Figure 1) that have trust relationships. They share information upon request and with cloud services provided in their subnetworks and the service provider's reputation information. Moreover, upon request, kingpins also share the reputation scores of service providers providing a specific service.

When a service consumer requires service, it can check whether any of the peers (located close by) are offering the service. In the case of non-availability of the required information locally, an entity can ask its peers for the service providers providing the required service. If it does not receive enough recommendations or the service provider reputation recommended by its peers is not up to the mark, the service consumer can request the kingpin node for which it registered for a service provider with a high reputation that provides the required service. Moreover, for better accuracy and the case that the service consumer entity is not resource-constrained, it can request the local kingpin node directly instead of relying on local knowledge collected from neighbours in its peers' range. The service consumer is responsible for rating the service provider after receiving service, even if the consumer is associated with another kingpin (e.g. fog-node B), and sharing the information with its domain kingpin, which in turn is held responsible for sharing the information with the kingpin B.

Moreover, in IoT, each entity can voluntarily join or leave the subnetwork. In such a scenario, after an entity has registered its services with the kingpin in the target subnetwork, it can provide the ID of the kingpin in the subnetwork to which it was previously registered. Kingpins in the target subnetwork can obtain the reputation of an entity from the cloud centre by providing information about the kingpin to which the node was previously registered.

After the expiry of every recent evaluation period, the IoT entities share the trust table containing the direct trust value for each service provider with which it interacted during the evaluation period. The kingpin receives the fine-grained values (representing direct trust) from the service consumers regarding a service provider in a specific context instead of sending feedback after every transaction. Let us suppose there are n service consumers who have taken service s_1 one or

more times from service provider j in the evaluation period ep . At the expiry of ep , each service consumer computes the recommendations (based on direct trust) and transmits them to the kingpin. The kingpins are held responsible for computing the current reputation of the service providers at the subnetwork (fog) level, based on the fine-grained information received. Instead of uploading all the data received from the nodes indiscriminately, only the fine-grained information, including the current reputation of the service providers and credibility of service consumers, is uploaded from fog-nodes to the cloud for further analysis. The cloud is held responsible for computing the global reputations of service providers and updating the credibilities of consumers. The cloud provides such global scores to the kingpins (fog-nodes) with which the IoT nodes are currently registered.

In the work presented in [10, 22, 24, 28, 37, 38], designated resource-full node in the subnetwork is responsible for the following: (i) providing trustworthiness value of the service provider, (ii) trustworthiness evaluation, (iii) providing list of nodes offering the specific service. Feedback data is pushed continuously to the central location. The feedback data arrival rate is sometimes high. Therefore, in this approach, a scalability issue arises due to increased bandwidth requirements. Moreover, in resource-constrained networks (such as WSNs), this approach may result in the network longevity issue and raise packet loss risks due to collisions.

In the case of resources constrained network, communication is the most expensive compared with computation. Therefore, rather than sending raw feedback to the kingpin node, finer-grained information should be returned by the nodes (service consumers) in the network. In this paper, each node must share with the kingpin the direct trust of the service provider computed based on the QoS received over the current evaluation period.

4 | THREAT MODEL

This paper deals with trust-related attacks that can impact the TRM system [32]. We made realistic assumptions about malicious behaviour of entities in the IoT environment (are in line with existing literature such as [10, 22, 24, 27, 28, 38]). We assume no link between a highly reputed service provider and its likelihood of being a target in the system. Instead, each service provider has the same likelihood of being targeted. Moreover, an entity with a good reputation as a service provider might not act as a good recommender. A malicious entity as a service provider can provide false information by lying about its reputation level. With a certain probability or for certain interactions, a misbehaving recommender entity may provide false recommendations randomly to increase or decrease the reputation of a service provider.

In addition, we allow malicious (misbehaving) service consumers to assign unfair ratings. Service providers may provide service with varying quality over time or periodically. The reasons for a change in behaviour might be malicious intent or a reduction in the number of resources (such as

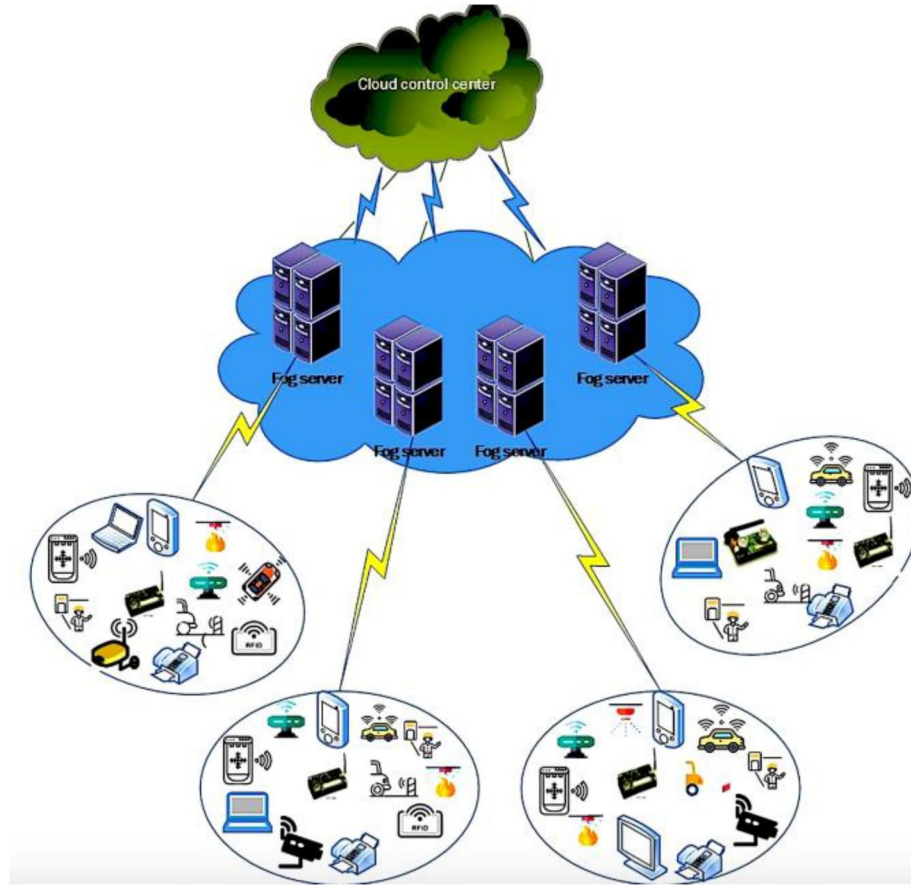


FIGURE 1 Kingpin (fog) nodes as subnetwork controllers

battery power). We considered the following five trust-related attacks that can disrupt the TRM system.

1. Self-promoting attack: A malicious service provider artificially promotes its importance by making good recommendations for itself as the service provider selection, but this can result in poor or faulty service after selection.
2. Bad-mouthing attack: A bad-mouthing attack occurs when dishonest service consumers provide bad recommendations to try to harm the reputation of a well-behaved service provider and reduce its chances of being chosen by other IoT entities. In addition, multiple malicious IoT entities (as recommenders) can collaborate to ruin the service trust-worthiness of the good service provider.
3. Ballot-stuffing attack: A ballot-stuffing attack occurs when service consumers try to boost the service provider's reputation by providing unfairly high recommendations. The aim is to increase the chances of its selection from other IoT entities for a specific service. In addition, multiple malicious IoT entities (as recommenders) can collaborate to boost the service trust of each other.
4. Opportunistic service attack: To deploy an attack once a reputation is gained, a malicious service provider builds a good reputation by having fair interactions initially. Afterwards, service providers abuse the earned reputation by

acting maliciously only on specific occasions. Service providers may provide service with varying quality over time or periodically or randomly. Moreover, with a good reputation score, the malicious service provider can effectively collude with other malicious entities to carry out attacks.

5. Discriminatory attack: A malicious entity discriminates against certain entities. While serving as a recommender, the victim service provider offers a bad service recommendation even if good service has been provided. On the other hand, for the non-target service providers, the recommender provides good service recommendations even if they provide bad service.

5 | TRUST MANAGEMENT PROTOCOL

In various applications, such as stock market data, WSNs feedback in the electronic communities, data may take the form of a stream of values [2]. In electronic communities, service consumers take services from a service provider over time, resulting in a stream of ratings (representing the QoS received from the service provider). The stream is unbound. A sliding bounded window is assumed to be defined across the data stream to deal with this problem. The evaluation domain is narrowed using the concept of windows. In this work, we

have considered the time-driven scheme to update the trust score. The trust score is updated after a specific period.

In our work, ratings given by the service consumer during direct interactions in the context C_x are stored in the recent time window represented by t_r . For time window t_r the dimension is of the form FROM *Start* TO *End*. The window size is represented using time units (such as HOURS). Let t represent the duration of the evaluation period. Let $\mathcal{T}_d(i, j, C_x, t_r)$ represents the direct trust computed by i for the service provider j , in the context C_x based on the ratings in t_r (computed using Equation 4). Direct trust score is updated (using Equation 1) after the expiry of t_r , representing direct trust on the service provider based on the previous and current behaviour.

Let t_0 represents the previous evaluation period at which the trustworthiness value of a service provider is updated. $\mathcal{T}_d(i, j, C_x, t_0)$ is updated after the expiry of t_r (Equation 1).

Entities may have limited memory, and therefore it is not possible to store every piece of feedback. Due to this constraint, they only store the entity reputations and the ratings given to service providers based on the QoS received in t_r . Initially (at the expiry of the first evaluation period), the service provider is assigned a default trustworthiness score [9] which is used as the value of $\mathcal{T}_d(i, j, C_x, t_0)$ in Equation (1). At the expiry of every t_r , service consumer i updates $\mathcal{T}_d(i, j, C_x, t_0)$ using Equation (1). $\mathcal{T}_d(i, j, C_x, t_r)$ is flushed to $\mathcal{T}_d(i, j, C_x, t_{r-1})$, and $\mathcal{T}_d(i, j, C_x, t_0)$ is updated Equation (1). The contents in t_r are flushed to make it ready for the next evaluation period:

$$\mathcal{T}_d(i, j, C_x, t_0) = \delta \mathcal{T}_d(i, j, C_x, t_0) + (1 - \delta) \mathcal{T}_d(i, j, C_x, t_{r-1}) \quad (1)$$

$\mathcal{T}_d(i, j, C_x, t_0)$ represents the direct trust score of the service provider based on its past behaviour. δ is a weighting factor, the value of which lies between 0 and 1. The weighting factor δ is used to give weightage to the direct trust computed at t_0 and t_{r-1} , while updating the direct trust of the service provider at t_0 .

5.1 | Direct trust calculation by IoT nodes

The QoS received from the service provider (in an interaction) is rated in the continuous range [0–1]. The rating represents the service consumer's level of satisfaction related to the service received from the service provider. The ratings given by the service consumer in the t_r boosts or reduces the corresponding trust value of the service providers accordingly. The direct trust is computed based on the contents in the recent time window (t_r). Using the concept of quantiles [46], we have divided the continuous rating range [0, 1] into equal-sized subsets to rank the ratings and recommendations provided by the entities.

The service consumer can rate the QoS received in t_r as Satisfied (S) or Dissatisfied (DS). S and DS can be classified as High (H), Medium (M), or Low (L) as shown in Figure 2. For highly satisfied service (S_H) the quality rating is assigned in the range (>0.83 and ≤ 1). Moreover, for service quality classified as medium satisfied (S_M), the rating can be

assigned in the range (≥ 0.67 and ≤ 0.83), and for low satisfied service (S_L), the quality is in the range S_L (≥ 0.5 and ≤ 0.67). For highly dissatisfied (DS_H), the service rating assigned lies in the range (≥ 0 and < 0.17); for medium dissatisfied (DS_M), the rating assigned lies in the range (≥ 0.17 and < 0.34); and for low dissatisfied (DS_L), the rating lies in the range (≥ 0.34 and < 0.5).

After the expiry of t , the outcome of the interactions with QoS classified as S ($O_S(i, j, C_x, t_r)$) and outcome of the interactions with DS ($O_{DS}(i, j, C_x, t_r)$) are calculated via Equations (2) and (3), respectively:

$$n_{S_H} + n_{S_M} + n_{S_L} = n_S$$

$$n_{DS_H} + n_{DS_M} + n_{DS_L} = n_{DS}$$

$$O_S(i, j, C_x, t_r) = \frac{\sum_{int=0}^{n_{S_H}} S_H(int) + \sum_{int=0}^{n_{S_M}} S_M(int) + \sum_{int=0}^{n_{S_L}} S_L(int)}{n_S} \quad (2)$$

$$O_{DS}(i, j, C_x, t_r) = \frac{\sum_{int=0}^{n_{DS_H}} DS_H(int) + \sum_{int=0}^{n_{DS_M}} DS_M(int) + \sum_{int=0}^{n_{DS_L}} DS_L(int)}{n_{DS}} \quad (3)$$

In this trust model, the context represents a specific type of service. $S_H(int)$ denotes that the QoS received in the interaction int (in t_r) is classified as S_H . n_{S_H} represents the total number of S_H based on the outcomes of interactions; n_{DS_H} is the total number of DS_H . n_S and n_{DS} represent the total number of S and DS interactions. The direct trust of i on j for context s_1 is calculated in Equation (4), where ρ is the weighting factor to weight $O_S(i, j, s_1, t_r)$ and $O_{DS}(i, j, s_1, t_r)$. ρ ranges between [0, 1]. $\mathcal{T}_d(i, j, C_x, t)$ represents the direct trust of service consumer i on the service provider j (5), computed based on its current ($\mathcal{T}_d(i, j, C_x, t_r)$) and past behaviour ($\mathcal{T}_d(i, j, C_x, t_0)$):

$$\mathcal{T}_d(i, j, s_1, t_r) = \rho O_S(i, j, s_1, t_r) + (1 - \rho) O_{DS}(i, j, s_1, t_r) \quad (4)$$

where $\rho + (1 - \rho) = 1$, and $\alpha + (1 - \alpha) = 1$.

$$\mathcal{T}_d(i, j, C_x, t) = \alpha \mathcal{T}_d(i, j, C_x, t_0) + (1 - \alpha) \mathcal{T}_d(i, j, C_x, t_r) \quad (5)$$

A node is autonomous in making decisions regarding how to calculate the reputation of a service provider considering the network conditions and other requirements. Consider an example scenario in which i has direct interactions greater than the threshold θ_D with j in t_r ; then, to meet the energy efficiency requirement, it can decide to

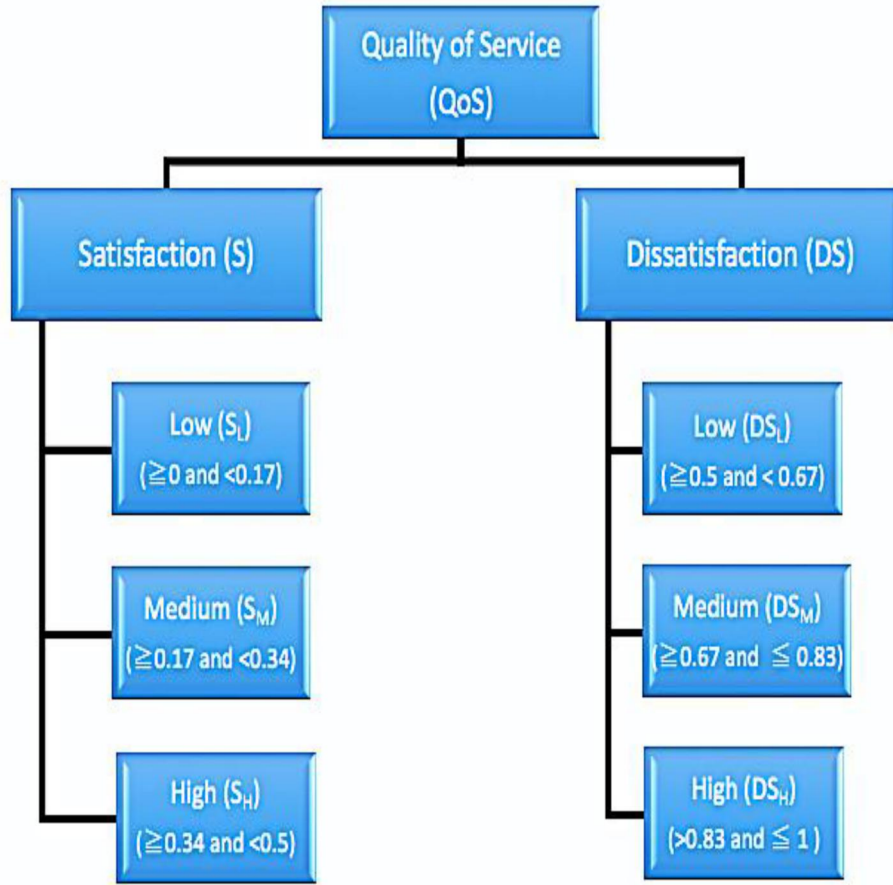


FIGURE 2 Continuous range $[0, 1]$ used for rating the received QoS from trustee is divided into equal-sized subsets

calculate the reputation based on the direct trust only. If i has fewer interactions than the threshold θ_D with j , then it can calculate the indirect trust on j by collecting recommendations from neighbours who have interacted with the j in the context C_x .

If the number of peers with good credibility scores is greater than θ_{ID} , i can calculate the indirect trust on service provider j based on the received recommendations. Afterwards, it can compute reputation based on direct and indirect trust (10). For better accuracy, or if the number of good credibility recommenders is less than θ_{ID} , then node i can query the kingpin node for the trustworthiness score of service provider j (17). Moreover, if a service consumer wants to obtain service from a service provider with a high reputation that is not available in its range or peers' range, it can ask the kingpin node.

5.2 | Indirect trust calculation by IoT nodes

The existing IoT-related works related to TRM systems consider time-sensitivity requirements and support giving different weights to the trustworthiness calculated for the recent period and previous period. In the existing literature related to TRM for IoT, the recommenders only provide the

single recommendation value computed based on current and past transactions upon request. Based on the reputation score provided, it does not reflect clearly whether the service provider is currently providing good-quality services or whether it has provided high-quality services in the past. Moreover, it also does not reflect whether the service provider is currently providing the service or not. Our trust framework allows the entities to update their belief sets for environmental changes. The scheme presented requires the recommender node to provide two trustworthiness scores of the service provider based on current performance and past performance.

Upon receiving a request for an indirect trust calculation from the service consumer i , the recommender node k is required to provide two scores of service provider j : one representing the direct trust on j computed based on the ratings provided to j in the recent time window ($R_d^{T(kj, C_x, t_r)}$) and the other representing its reputation (computed based on its direct and indirect trust) ($R^{T(ij, C_x, t)}$) updated during the last evaluation period. Our trust mechanism requires recommenders to send the service provider's recent direct trust score ($R_d^{T(kj, C_x, t_r)}$) to monitor the service provider's current behaviour, meet adaptivity requirements, and address TRM system attacks (such as opportunistic and discriminatory attacks).

5.2.1 | Credibility computation for recommender entities

Our scheme incorporates a mechanism to compute recommender entity credibility and achieve survivability against trust-based attacks such as bad-mouthing and ballot-stuffing. The credibility of the malicious recommender is decreased with each malicious attempt, which allows converging the trustworthiness of the service provider to the ground truth in the presence of malicious nodes. The goal of malicious peers is to cause harm to specific members of the network or to the entire system. Upon receiving direct trust value ($R^{T_d(k,j,C_x,t_r)}$) (part of recommendation) from a recommender k about the service provider j ; rank is assigned to the received direct trust value according to Table 1. For example, if for j , k provides the current direct trust value 0.83, then it will be ranked as 5.

By default, the credibility of a recommender node is 0.5. Recommender credibility is computed and maintained by the service consumer to determine the trustworthiness of the recommendation provided by the recommender. The credibility of the recommender helps in addressing bad-mouthing and ballot-stuffing attacks. The credibility of an entity as a recommendation source is updated based on its recent behaviour as a recommender. The credibility of a malicious recommender is decreased with each malicious attempt, which allows for the convergence of the trustworthiness of the service provider to the ground truth in the presence of malicious nodes.

To compute the credibility of a peer k , k 's direct trust (based on tuples in t_r) on service provider j is compared with (i) the service consumer i direct trust (based on tuples in t_r) on service provider j , and (ii) also with the average of direct trust scores received from other peers representing the QoS provided by j to them at t_r in the context C_x . If the rating behaviour is similar (in the acceptable range) to that from the service consumer or to the average of the direct trust scores provided by the peers who rated the service consumer for a similar context and period, then the credibility of the recommender is increased by some points. Otherwise, points are deducted from the credibility as a forfeiture payment. The points to be credited or debited from the credibility of the recommender depend on the extent of the rating behavioural difference.

TABLE 1 Recommendation and direct trust score rank table

Rank	Range
1	(≥ 0 and < 0.17)
2	(≥ 0.17 and < 0.34)
3	(≥ 0.34 and < 0.5)
4	(> 0.83 and ≤ 1)
5	(≥ 0.67 and ≤ 0.83)
6	(≥ 0.5 and ≤ 0.67)

Let us consider a scenario. Service consumer i has computed the direct trust value $T_d(i,j,s_1,t_r)$ for service provider j based on direct interactions taking service s_1 . To compute the indirect trust related to j for context s_1 , it takes recommendations from nodes m ($\{R^{T_d(m,j,C_x,t_r)}, R^{T(m,j,C_x,t)}\}$), a ($\{R^{T_d(a,j,C_x,t_r)}, R^{T(a,j,C_x,t)}\}$), b ($\{R^{T_d(b,j,C_x,t_r)}, R^{T(b,j,C_x,t)}\}$), c ($\{R^{T_d(c,j,C_x,t_r)}, R^{T(c,j,C_x,t)}\}$), and d ($\{R^{T_d(d,j,C_x,t_r)}, R^{T(d,j,C_x,t)}\}$). Upon receiving recommendations from entities m , a , b , c , and d , entity i will update the credibility of m , a , b , c , and d . Upon receiving ($R^{T_d(m,j,C_x,t_r)}, R^{T_d(a,j,C_x,t_r)}, R^{T_d(b,j,C_x,t_r)}, R^{T_d(c,j,C_x,t_r)}, R^{T_d(d,j,C_x,t_r)}$) ranking will be assigned as given in Table 1. e.g. if $R^{T_d(a,j,C_x,t_r)}$ lies in the range (> 0.83 and ≤ 1), it will be ranked as 4. The credibility updating of recommender a by service consumer i is performed as follows.

Step 1 Upon receiving ($R^{T_d(a,j,C_x,t_r)}$) from a as a recommendation, service consumer i computes the absolute difference between the ranking categories, where the direct trust value given by i and a is represented by $RS_{i,a} = \text{ABS}(\text{Diff}(\text{Rank}(T_d(i,j,s_1,t_r)), \text{Rank}(R^{T_d(a,j,C_x,t_r)})))$. Ranking categories are presented in Table 1.

Based on the RS of a with i ($RS_{i,a}$), the credibility of recommender a is credited or debited with some points. If the difference between the rating categories represented by $RS_{i,a}$ is zero, the credibility of the recommender a will be increased with a threshold (ζ_1). If the $RS_{i,j}$ is 1, a credibility will be decreased with ζ_2 points. If the $RS_{i,j}$ is 2, a credibility will be decreased by ζ_3 points. If the $RS_{i,j}$ is 3, it will be decreased by ζ_4 points. Otherwise, with a threshold ζ_5 . $\text{RSTrustor}(T_d(i,j,c_x,t_r), R^{T_d(a,j,C_x,t_r)})$ in Equation (7) represents the RS between i and a .

Step 2 Service consumer node i computes the average ($uR^{T_d(j,C_x,t_r)}$) of all the recommendations (direct trust scores only) received from neighbour recommenders (except j and a) for service provider j in the context C_x (Equation 6). To address self-promoting attacks, the service provider j recommendation for itself is not considered. Furthermore, to deal with a discriminatory attack recommendation provided by the recommender a is also not considered, while computing the mean of recommendations (direct trust scores only) made by the peers of i . To address self-promoting attacks, self-recommendations by service providers are not considered:

$$uR^{T_d(j,C_x,t_r)} = \frac{\sum_{k \in N_i, k \neq a, k \neq j} R^{T_d(k,j,C_x,t_r)}}{n_{R^{T_d(i,j,C_x,t_r)}}} \quad (6)$$

Service consumer node i then computes the absolute difference between the ranking categories where the recommendation (direct trust) given by a and $uR^{T_d(j,C_x,t_r)}$ lie, represented

by $RS_{a,uR^{T_d(j,C_x,t_r)}} = (ABS(diff(Rank(R^{T_d(a,j,C_x,t_r)}), Rank(uR^{T_d(j,C_x,t_r)})))$.

Based on the rating behaviour similarity ($RS_{a,uR^{T_d(j,C_x,t_r)}}$), the credibility of recommender a is credited or debited with some points (represented by $(RS_PeersAVG(R^{T_d(a,j,C_x,t_r)}, uR^{T_d(j,C_x,t_r)}))$ in Equation 7). If the difference between the rating categories is zero, the credibility of recommender a will be increased with a certain threshold of (ζ_1) points. If the difference is 1, it will be decreased by ζ_2 points. If the difference is 2, it will be decreased by ζ_3 . If the difference between categories is 3, the credibility of recommender a will be decreased by ζ_4 points. Otherwise, it will be decreased with a threshold ζ_5 .

The credibility of the recommender a is updated as shown in Equation (7). The value of τ lies between 0 and 1. If the trustworthiness value based on direct trust is not computed in the current time window, the value of τ is assumed to be zero:

$$Cr_a = \tau RSTrustor(T_d(i, j, C_x, t_r), R^{T_d(a,j,C_x,t_r)}) + (1 - \tau) RS_PeersAVG(R^{T_d(a,j,C_x,t_r)}, uR^{T_d(j,C_x,t_r)}) \quad (7)$$

5.2.2 | Indirect trust computation by IoT entities

In Equation (7), $T_{n_Nd}(i, j, C_x, t_r)$ represents the indirect trust of service consumer i on service provider j for the context C_x computed based on local knowledge of node (an entity). $T_{n_Nd}(i, j, C_x, t_0)$ represents the indirect trust updated on the previous evaluation period, stored locally at i (representing the j past indirect trust score). $T_{n_Nd}(i, j, C_x, t_r)$ is computed by i based on the reputation scores provided by the recommending peers in the recent evaluation period (representing j current reputation). N_i represents the neighbourhood of i (i.e. peers that are neighbours of node i). $k \in N_i$ represents that entity k belongs to the neighbourhood of i , and Cr_k represents the credibility of entity k . The credibility of the recommender lies between 0 and 1. To meet the adaptivity requirement and to address opportunistic attack, our trust model considers the service provider past and current behaviour. To address self-promoting attacks, self-recommendations provided by service providers are not considered $k \neq j$ in Equation (8):

$$T_{n_Nd}(i, j, C_x, t_r) = \frac{\sum_{k \in N_i, k \neq j} R^{T_d(k,j,C_x,t_r)} * Cr_k}{\sum_{k \in N_i, k \neq j} Cr_k} \quad (8)$$

Equation (9) represents the indirect trust calculated at time t with support from neighbouring nodes. ω is a weighting factor used to weight past indirect trustworthiness and recent indirect trustworthiness, the value of which ranges between [0–1]. $T_{n_Nd}(i, j, C_x, t)$ represents the indirect trust of the

service provider j , computed based on its current ($T_{n_Nd}(i, j, C_x, t_r)$) and past behaviour ($T_{n_Nd}(i, j, C_x, t_0)$) (Equation 9). To address the time-sensitivity requirement, the past and present behaviour of service providers is considered:

$$T_{n_Nd}(i, j, C_x, t) = \omega T_{n_Nd}(i, j, C_x, t_0) + (1 - \omega) T_{n_Nd}(i, j, C_x, t_r) \quad (9)$$

5.3 | Reputation computation by an IoT entity based on local knowledge only

The reputation of the service provider should be based on current and past behaviour. If more weight is given to past behaviour relative to current behaviour or the present behaviour is not considered, then calculated trustworthiness would not be accurate.

The reputation ($T(i, j, C_x, t)$) of service provider j is computed by service consumer i based on local knowledge only (in Equation 10) using direct ($T_d(i, j, C_x, t)$) and indirect trust ($T_{n_Nd}(i, j, C_x, t)$). β is used as a weighting factor to weight direct and indirect trust. The lower value of β means giving more weight to indirect trust:

$$T(i, j, C_x, t) = \beta T_d(i, j, C_x, t) + (1 - \beta) T_{n_Nd}(i, j, C_x, t) \quad (10)$$

5.4 | Global trust computation by kingpin

After the expiry of every recent evaluation period, the IoT entities share the trust table containing the direct trust value for each service provider with which they interacted during the evaluation period. The kingpin receives the fine-grained values (representing direct trust) from the entities who have taken any service from any other entity (as a service provider) in the IoT network. Let us suppose there are n service consumers in the network who have taken service s_1 one or more times from service provider j during the current (t_r) time window. At the expiry of t_r , each service consumer computes the direct trust using Equation (4) and transmits it to the kingpin (fog-node).

At a kingpin, the direct trust values received in the context C_x are stored in the recent time window represented by t_r^{kp} . For time window t_r^{kp} the dimension is of the form FROM Start TO End. The window size is represented using time units (such as HOURS). Let t^{kp} represent the time at which the t_r^{kp} expires, upon which the kingpin evaluates the service provider trustworthiness score.

Moreover, t_r^{kp} defines the recent time window that stores the recommendations $[T_d(1, j, s_1, t_r) \dots T_d(n, j, s_1, t_r)]$ received for the service provider j in the context s_1 . Initially, at the first evaluation, a default reputation score in the context s_1 is assigned

to the service provider, which is used as the value of $\mathcal{T}_{kp_nd}(j, s_1, t_0^{kp})$. At the expiry of t_r^{kp} , $\mathcal{T}_{kp_nd}(j, s_1, t)$ is assigned to $\mathcal{T}_{Nd}(i, j, C_x, t_0)$. The updated $\mathcal{T}_{kp_nd}(j, s_1, t)$ value is computed as shown in Equation (13).

At the expiry of t_r^{kp} , the kingpin checks the dispersion in the n recommendations received for a service provider j in the context s_1 . If there is high dispersion in the received data $[\mathcal{T}_d(1, j, s_1, t_r^{kp}), \dots, \mathcal{T}_d(n, j, s_1, t_r^{kp})]$ (i.e. $\sigma = 1.44 * \text{Median Absolute Deviation (MAD)}$ [47, 48] of these values is greater than threshold ψ), outliers are removed (leaving p recommendations). MAD is more resilient to outliers in a data set than the standard deviation.

We apply ABS ($\text{Median} \pm k(\sigma_MAD_R)$) to detect the outliers. The credibility of the recommender a $Cx_a^{kp_nd}$, who provided such extreme recommendation, is reduced with more points (decreased by ζ_5). The credibility of other recommenders is updated based on the RS difference between the recommender a with the average of recommendations provided by the p recommenders in the t_r^{kp} , for service provider j in the context s_1 (Equation 11). After removing the extreme values, the kingpin k computes the average ($u\mathcal{T}_d(j, s_1, t_r^{kp})$) of all the remaining p recommendations received from recommenders for service provider j in the context s_1 (see Equation 11).

As summarized in Table 1, kingpin node k computes ranking categories of recommendation given by recommender a , and $u\mathcal{T}_d(j, s_1, t_r^{kp})$. Kingpin node k then computes the absolute ranking similarity difference between these ranking categories (Equation 12). If the difference between the rating categories is zero, the credibility of a will be increased with a certain threshold (ζ_1). If the difference is 1, the credibility of a will be decreased with ζ_2 . If the difference is 2, it will be decreased with ζ_3 . If the difference between categories is 3, the credibility of a will be decreased with ζ_4 . Otherwise, it will be decreased with a threshold ζ_5 .

Upon inquiry by the service consumer for the specific service provider j , the kingpin will provide the trustworthiness values of j , representing its trustworthiness at the expiry of t_0^{kp} and at t_r^{kp} . Global trust ($\mathcal{T}_{kp_nd}(j, s_1, t)$) is calculated using Equation (13) by the kingpin node. To address self-promoting attack recommendation provided by the service provider for itself is not considered ($i \neq j$ in Equation 11):

$$\mathcal{T}_{kp_nd}(j, s_1, t_r^{kp}) = \frac{\sum_{i=1, i \neq j}^{i=p} \mathcal{T}_d(i, j, s_1, t_r) * Cx_i^{kp_nd}}{\sum_{i=1}^{i=p} Cx_i^{kp_nd}} \quad (11)$$

$$Cx_j^{kp_nd} = \text{RS_PeersAVG}(\mathcal{T}_d(a, j, s_1, t_r^{kp}), u\mathcal{T}_d(j, s_1, t_r^{kp})) \quad (12)$$

$$\mathcal{T}_{kp_nd}(j, s_1, t) = \alpha \mathcal{T}_{kp_nd}(j, s_1, t_0^{kp}) + (1 - \alpha) \mathcal{T}_{kp_nd}(j, s_1, t_r^{kp}) \quad (13)$$

5.5 | Global trust computation by an IoT entity based on local and global knowledge

As we discussed in Section 3, if a service consumer does not find enough recommendations for a service provider from credible neighbours, then what will happen? The service consumer will immediately ask for the reputation of the service provider from the kingpin. Even for better accuracy in computing the reputation of the service provider, it can query the kingpin node. At last, the reputation of service provider j is computed by node i (using direct and indirect trust) in Equation (17). Indirect trust is computed based on local and global knowledge (see Equation 16). β is used as a weighting factor to weight direct and indirect trust. The lower value of β means giving more weight to indirect trust.

Based on the service consumer accuracy requirement, the indirect trust for a service provider can be computed with the support of both neighbouring nodes and the kingpin (using Equation 16). For indirect trust (\mathcal{T}_{Nd}) the calculation is based on the information received from kingpin (\mathcal{T}_{kp_nd}) and indirect trust computed by the consumer node itself (\mathcal{T}_{n_nd}) for the period t_0 , and t_r (8). $\mathcal{T}_{Nd}(i, j, C_x, t)$ represents the indirect trust calculated based on the past ($\mathcal{T}_{Nd}(i, j, C_x, t_0)$ in Equation 14) and current behaviour ($\mathcal{T}_{Nd}(i, j, C_x, t_0)$ in Equation 15).

$$\mathcal{T}_{Nd}(i, j, C_x, t_0) = \lambda * \mathcal{T}_{n_nd}(i, j, C_x, t_0) + (1 - \lambda) * \mathcal{T}_{kp_nd}(j, C_x, t_0^{kp}) \quad (14)$$

$$\mathcal{T}_{Nd}(i, j, C_x, t_r) = \lambda * \mathcal{T}_{n_nd}(i, j, C_x, t_r) + (1 - \lambda) * \mathcal{T}_{kp_nd}(j, C_x, t_r^{kp}) \quad (15)$$

In Equations (14) and (15), λ represents the weighting factor to weight the indirect trust computed based on the local knowledge (\mathcal{T}_{n_nd}) and the indirect trust score provided by the kingpin (\mathcal{T}_{kp_nd}). The value of λ , ranges between [0,1]:

$$\mathcal{T}_{Nd}(i, j, C_x, t) = \alpha * \mathcal{T}_{Nd}(i, j, C_x, t_0) + (1 - \alpha) * \mathcal{T}_{Nd}(i, j, C_x, t_r) \quad (16)$$

$$\mathcal{T}(i, j, C_x, t) = \beta \mathcal{T}_d(i, j, C_x, t) + (1 - \beta) \mathcal{T}_{Nd}(i, j, C_x, t) \quad (17)$$

6 | EXPERIMENTAL EVALUATION

This section demonstrates that our proposed TRM protocol addresses desirable convergence, adaptivity, survivability, as well as resilience against malicious attacks requirements. The experiments are performed on 50 nodes that are deployed randomly. A discrete-event network simulator Network

Simulator 3 (NS-3) [49, 50], is selected as the simulator. The entities (nodes) interaction pattern follows the distribution supported by the analysis of real traces specified in [9, 20, 51, 52]. The service consumer can rate the service provider between 0 and 1.

In this work, several experiments have been conducted, and impacts of different weighting factors have been shown on the trustworthiness of nodes. A comparison of the work has been made with the current work [9–11, 22]. In the first three experiments (Sections 6.1, 6.2 and 6.3), we demonstrate that the proposed TRM protocol addresses the adaptivity requirement. Afterwards, we demonstrate the survivability property of the TRM protocol (Section 6.4). We then demonstrate the scalability property of our TRM protocol. Lastly, in the last two experiments, we show the network resilience against two trust-related attacks (bad-mouthing and ballot-stuffing) with different percentages of malicious nodes.

Among existing works (related to the TRM system for IoT and SIoT), only [10] provides experimental evidence related to addressing the survivability requirement.

In [9, 20], experimental evidence is provided related to addressing adaptive behaviour and fast convergence requirements. The work in [20] assumes that friends are cooperative, and the cooperativeness value is computed as the ratio of the number of common friends and the total number of friends. Moreover, the social relationship between the owners of the entities in a transaction is exploited to measure the trustworthiness between entities. However, entities are able to build their own social network, which may be different and independent of their owners' social network. Static values are considered for representing relationships. Therefore, for meaningful comparison, we considered [9]. A fog-based trust scheme is presented in [22], therefore for meaningful comparison related to addressing scalability requirement, we selected it.

6.1 | Experiment 1: New user achieves ground truth based on direct trust evaluation

In this experiment, we show that a new entity can build its trustworthiness towards the service provider(s) with desirable convergence behaviour. The experiment is conducted to demonstrate that the TRM protocol addresses the adaptivity requirement. The aim of this experiment is to show the effect of trust parameter α (in Equation 5) on trust evaluation results based on interactions between a service consumer and a service provider that are randomly picked. To compare the proposed scheme with the work in [9], a similar experiment is conducted. The experimental setting is the same as that of [9]. The entities (nodes) interaction pattern follows the distribution supported by the analysis of real traces specified in [9, 20, 51, 52]. The initial direct trust of all nodes is set to ignorance (0.5). The average interaction frequency is approximately six times per day. Therefore, the recent time window size (t_r) is set to 2 days. When two legitimate service consumers interact directly with a service provider, the rating assigned may vary. For malicious service

consumer, the deviation from the ground truth is higher. The value of α is updated by choosing three values, 0.1, 0.3, and 0.9. Moreover, when α is less than 0.5, more weight will be given to the direct trust computed based on recent time window ($\mathcal{T}_d(i, j, C_x, t_r)$); when α is greater than 0.5, more weight will be given to the direct trust updated in t_0 ($\mathcal{T}_d(i, j, C_x, t_0)$); and when $\alpha = 0.5$, the same weight will be given to $\mathcal{T}_d(i, j, C_x, t_r)$ and $\mathcal{T}_d(i, j, C_x, t_0)$. Table 2 shows the configuration parameters for this experiment.

In Figure 3, the horizontal straight line indicates the actual trust value derived from the ground truth. When the value of α increases (i.e. more weight is given to the recent time window), the trust convergence time becomes shorter. Figure 3 shows that the presented scheme converges to ground truth more quickly as compared to [9]. This experiment demonstrates that giving more weight to direct trust computed by the service consumer, based on recent interactions, will allow correct learning of the behaviour of service providers.

6.2 | Experiment 2: Adaptive behaviour of recommenders

The experiment is conducted to demonstrate that the TRM protocol addresses the adaptivity requirement. This experiment shows how well our proposed scheme identifies the discriminatory behaviour of a recommender (discriminatory attack) and is equipped with the mechanism to address it. This experiment shows the change in 10 recommender entities' credibilities with different proportions of malicious entities. The X -axis represents the entity (node) number, and the Y -axis represents the proportion of malicious entities. All the recommender entities are providing recommendations to service consumer with identity 11. The experiment is based on indirect trust. In each round, the proportion of malicious entities increases. Initially, the proportion of malicious entities is 0.1, and therefore node 1 is set to be the malicious node. Afterwards, the proportion of malicious nodes is increased to 0.2, and therefore node 1 remained malicious and 2 is set to malicious. Finally, the proportion of malicious nodes is increased to 0.5, nodes 1 to 4 remains malicious, and node 5 is set to malicious. In all rounds, nodes 6 through 10 exhibited good behaviour. The good recommender nodes provide

TABLE 2 Simulation configuration parameters for Experiment 1

Parameters	Value
Initial trust value of nodes	0.5
Total number of nodes	40
Malicious nodes	20%
Fair rating	0.83–0.9
Dishonest rating	0
α	0.1, 0.3, 0.9
β	0

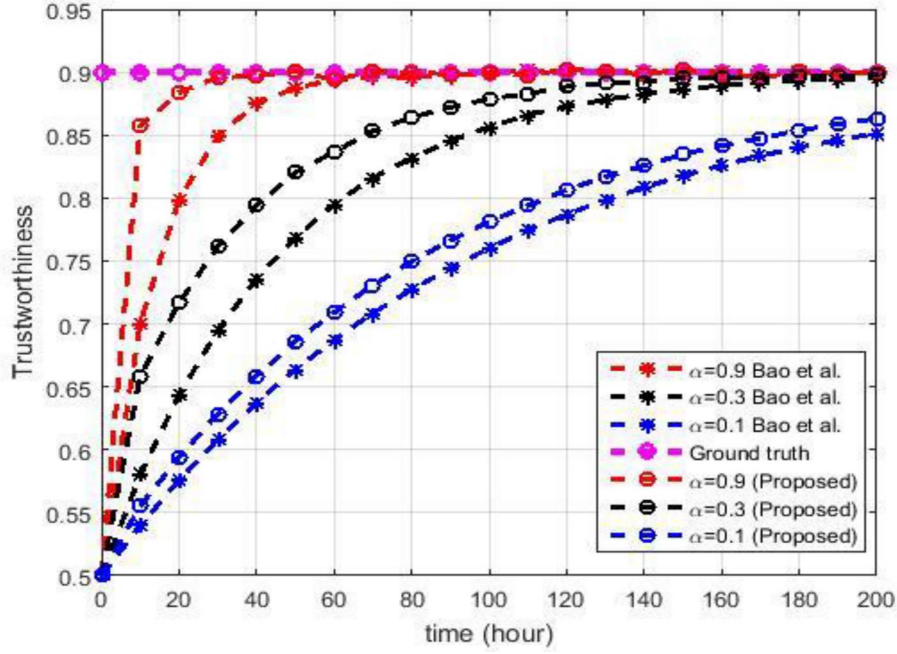


FIGURE 3 Effect of trust parameter α on direct trust evaluation (Comparison of our scheme with [9])

trustworthiness between $[0.83, 0.9]$ and malicious recommender nodes in the range $[0, 0.2]$.

The trustworthiness of the entity as a recommendation source (referred to as credibility) is an important parameter in a trust model. The credibility of the malicious recommender is decreased with each malicious attempt, which allows the convergence of service provider trustworthiness to the ground truth in the presence of malicious nodes. The values of the parameters required for computing recommender k 's credibility are as follows: ζ_1 is set to +5%, ζ_2 to -5%, ζ_3 to -10%, and ζ_4 to -15%. Considering Table 1, if the difference between the rating categories is zero, the credibility of recommender k will be increased within a certain threshold (5%) (considering Equation 7). If the difference is 1, it will be decreased by -5%. If the difference is 2, it will be decreased by -10%. If the difference between categories is 3, the credibility of recommender k will be decreased by -15% points.

Figure 4 shows the results using our proposed scheme. It is obvious to see that when a recommender entity provides a false recommendation, there will be a decrease in its credibility, and the credibility of a good behaviour node will increase in every round. In our presented scheme, the credibility of a node is decreased based on the amount of RS difference.

6.3 | Experiment 3: Addressing adaptive behaviour of the service provider (opportunistic service attack)

The experiment is conducted to demonstrate that the TRM protocol addresses the adaptivity requirement. Like the work in [11], a set of experiments is conducted to demonstrate that our

proposed scheme identifies the changing behaviour of a service provider (opportunistic service attack) and is equipped with the mechanism to address it. Our scheme allows the service consumer to select a service provider based on current behaviour or on a combination of both past and present. To meet the adaptivity requirement and to address opportunistic attack, our trust model considers the service provider past and current behaviour. This helps service consumers determine the change in service provider behaviour if modified.

The service consumer has a maximum of 40 neighbours. We assume that their interaction pattern follows the distribution supported by the analysis of many real traces [9, 20, 51, 52]. For two nodes, we consider that the average inter-contact interaction frequency is approximately six times per day. Therefore, the recent time window size (t_r) is set to six interactions, and the duration of t_r is set to 1 day.

The trustworthiness of the service provider is computed based on direct trust. In this set of experiments, alpha (α) is used as a weighting factor that weights the previous direct trust value computed at the expiry of the previous time window (t_0), and recent direct trust value computed based on the contents in the recent time window (t_r). Moreover, when α is less than 0.5, more weight will be given to the direct trust computed based on recent time window ($\mathcal{T}_d(i, j, C_x, t_r)$); when α is greater than 0.5, more weight will be given to the direct trust updated in t_0 ($\mathcal{T}_d(i, j, C_x, t_0)$); and when $\alpha = 0.5$, the same weight will be given to $\mathcal{T}_d(i, j, C_x, t_r)$ and $\mathcal{T}_d(i, j, C_x, t_0)$. Figure 5a shows the service provider milking behaviour. In this scenario, a node first builds its reputation, and after a period of time (making its reputation), it provides low QoS due to malicious intent or because of a lack of resources. Up to the 25th day, the outcome of the interaction is highly satisfactory (the service provider is providing good

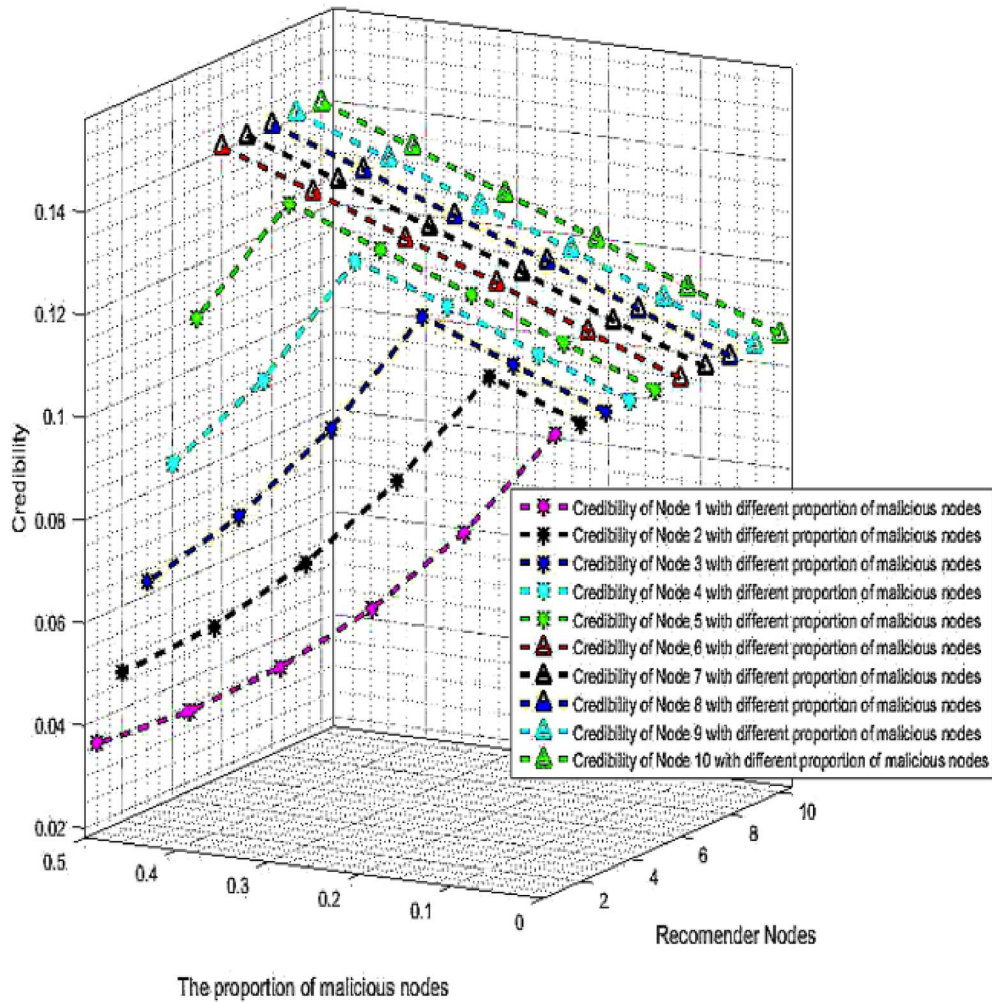


FIGURE 4 Change in recommender node credibility with different proportions of malicious nodes

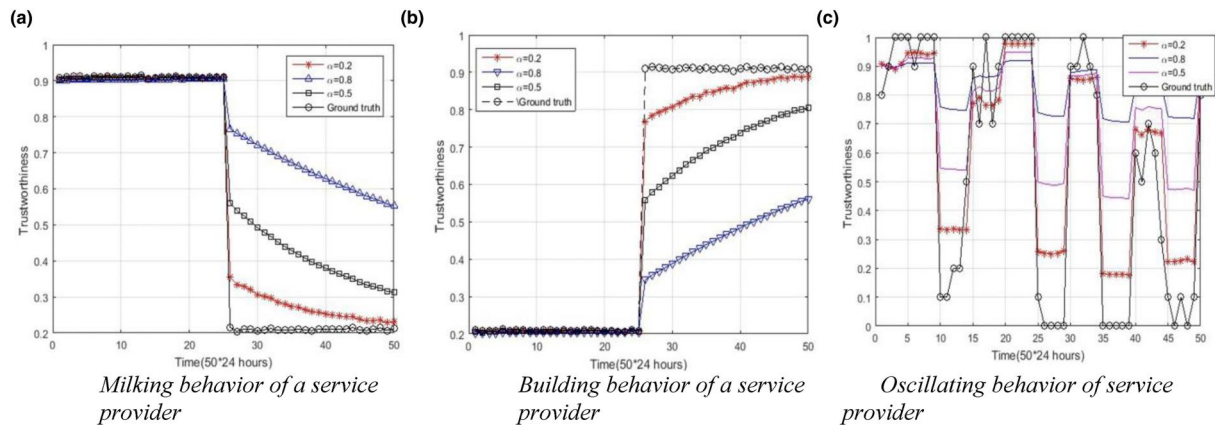


FIGURE 5 Addressing adaptive behaviour of the service provider (Opportunistic service attack). (a) Milking behaviour of a service provider, (b) Building behaviour of a service provider and (c) Oscillating behaviour of service provider

QoS, i.e. between $[0.88, 0.92]$). After the 25th day, the outcome of the interaction is dissatisfactory—that is, the QoS provided by the service provider lies in the range $[0, 0.2]$. Figure 5b shows that with the passage of time, the service provider

improves its service quality. Up to the 25th day, the outcome of the interaction is dissatisfactory and lies in the range $[0, 0.2]$, and after the 25th day, the outcome of the interaction is highly satisfactory, that is, between $[0.88, 0.92]$. For accuracy, it

necessitates consideration of both the past and recent behaviour of the service provider while calculating the trustworthiness of a service provider. The results show that the proposed scheme quickly adapts to the changing behaviour of the service provider.

Figure 5c shows the oscillating behaviour of a service provider separately. With this type of behaviour, a node behaves bad and good, performing badly for 10 days and then good for 10 days. In all experiments, the peers are required to provide recommendations based on current and past experience. It can be seen from Figure 5a,b and c; our scheme is able to detect the changing behaviour of service provider with the help of implementing the concept of time windows.

6.4 | Experiment 4: Achieving survivability by removing the impact of malicious users on the trustworthiness of the service provider

In this experiment, among 40 recommender nodes, 20% show malicious behaviour regarding a well-behaving service provider. Simulation configuration parameters are set as specified in [10]. The proposed protocol provides protection against this attack by building and updating the credibility of recommender nodes.

In [10], the credibility of a recommender entity is updated based on the quality of recommendation scores. When a malicious entity regularly provides unfair recommendations, its credibility is continuously decreased and finally becomes zero, and that any recommendation provided by an entity whose credibility is zero will be weighted with zero.

As shown in Figure 6, without considering the credibility, the trust level of an honest service provider is badly affected. The service provider trust level is initially decreased, but the system quickly recovers its trustworthiness by reducing the credibility of malicious nodes. It shows the comparison of our approach with [10]. Table 3 shows the configuration parameters for this experiment. It can be seen from Figure 6 that our approach impacts the reputation of a service provider in the presence of malicious users less adversely compared to the work presented in [10]. Furthermore, it allows for quick convergence to ground truth by reducing the credibility of recommenders that perform maliciously over a certain period of times. The results show that the trustworthiness of the service provider converges to the ground truth with the passage of time in the presence of malicious nodes. Figure 6, demonstrates that if we give more weightage (β) to direct trust (considering Equation 10) compared to indirect trust then it

TABLE 3 Simulation configuration parameters for Experiment 4

Parameters	Value
Initial credibility of recommender nodes	1
Total nodes	40
Malicious recommender nodes	20%, 8 nodes
Fair rating	1
False rating	0
Alpha (proposed)/Theta [10]	0.7
Lemda [10]	0.5

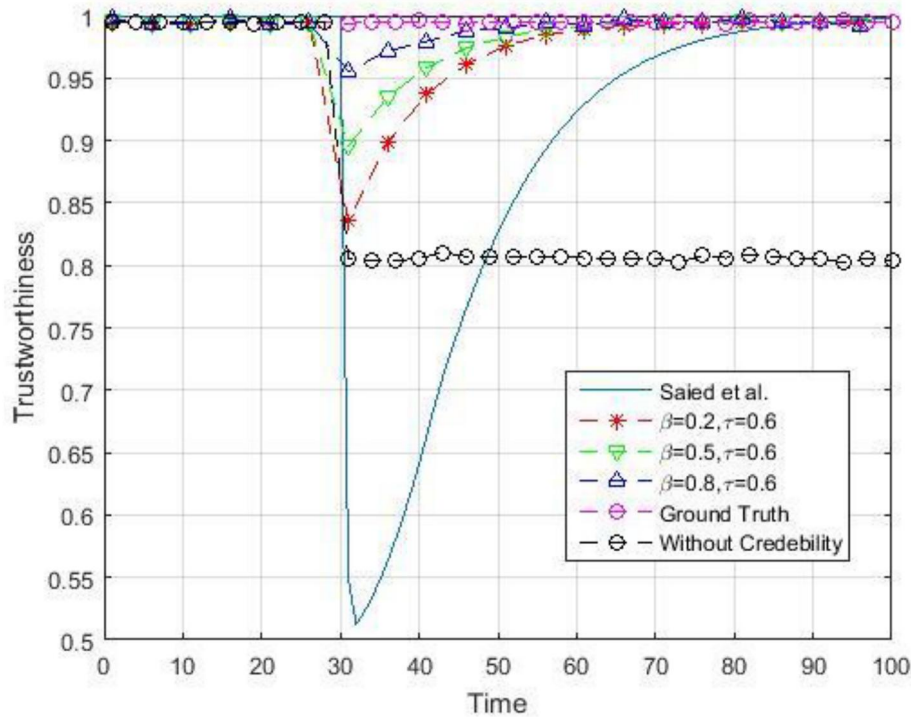


FIGURE 6 Achieving survivability: Removing the impact of malicious users on the trustworthiness of the service provider (comparison with [10])

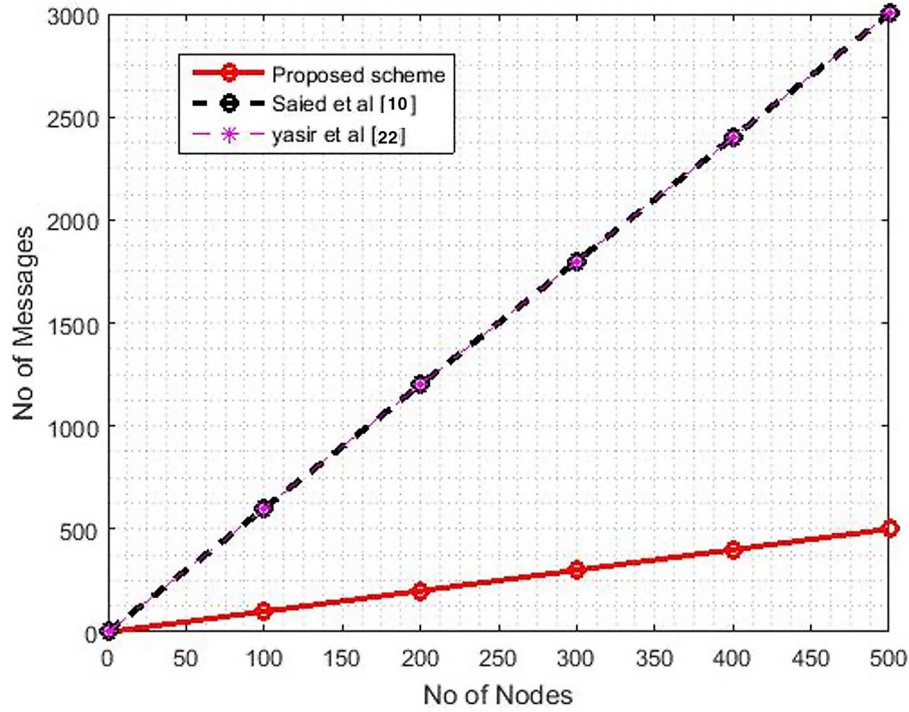


FIGURE 7 Communication overhead versus network size

allows quick convergence to the ground truth. The weight ($\tau = 0.6$) represents that while computing the credibility of recommender k , more weightage is given to k 's rating behaviour similarity with the service consumer i in context C_x (Equation 7). Less weightage ($\tau = 0.4$) is given to recommender k rating behaviour similarity with the average ($uR_d(j, C_x, t_r)$) of all the recommendations received from neighbour recommenders for service provider j in the context C_x (in Equation 7). The value of the parameters required for computing credibility are set as follows: ζ_1 is set to +5%, ζ_2 to -5%, ζ_3 to -10%, and ζ_4 to -15%. Considering Table 1, if the difference between the rating categories is zero, the credibility of recommender k will be increased within a certain threshold (5%) points (in Equation 7). If the difference is 1, it will be decreased by -5%. If the difference is 2, it will be decreased by -10%. If the difference between categories is 3, the credibility of recommender k will be decreased by -15% points. Otherwise, will be decreased with a threshold of -20%.

6.5 | Experiment 5: Addressing scalability requirement

In this experiment, we demonstrate the scalability property of our proposed TRM protocol. In this experiment, we assume that a network node takes services from a service provider in a similar context. Upon taking service, the node gives a rating about a service provider based on the QoS received. The nodes interaction pattern follows the distribution supported by the analysis of real traces specified in [9, 20, 51, 52]. The initial direct trust of

all nodes is set to ignorance (0.5). The average interaction frequency is approximately six times per day. Therefore, the recent time window size (t_r) is set to 2 days. In this work, we have considered the time-driven scheme to update the trust score. A temporal data reduction scheme is presented to reduce the total amount of data uploaded to the fog servers. At the expiry of t_r , each service consumer node computes the direct trust based on the ratings stored in the t_r and sends it to the fog server (kingpin). In contrast, in [10, 22] information (rating given to service provider after an interaction) is sent to the central location without in-network processing. In [10, 22], each interaction rating is sent to the central location (cloud), whereas in [22], each interaction rating is sent to the fog server. It can be seen in Figure 7 that because of in-network processing, our scheme generates less overhead relative to [10, 22].

6.6 | Experiment 6: Addressing bad-mouthing attack

Figure 8 illustrates the effect of bad-mouthing on the trustworthiness of a service provider with different percentages of malicious nodes. This experiment is conducted over a network of 50 nodes. The experiment shows the results of bad-mouthing attack on a single service provider having a peer range of 40 nodes. All the 40 service consumers are providing recommendations about a service provider in a similar context. We compared our results with the average of recommendations provided by non-malicious users (termed as normal trust in Figure 8a,b and c).

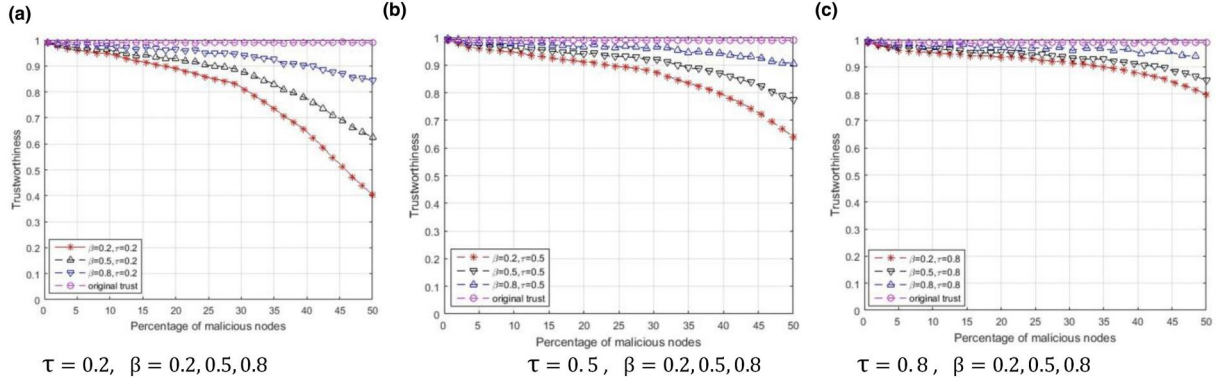


FIGURE 8 Addressing bad-mouthing attack. (a) $\tau = 0.2, \beta = 0.2, 0.5, 0.8$, (b) $\tau = 0.5, \beta = 0.2, 0.5, 0.8$ and (c) $\tau = 0.8, \beta = 0.2, 0.5, 0.8$

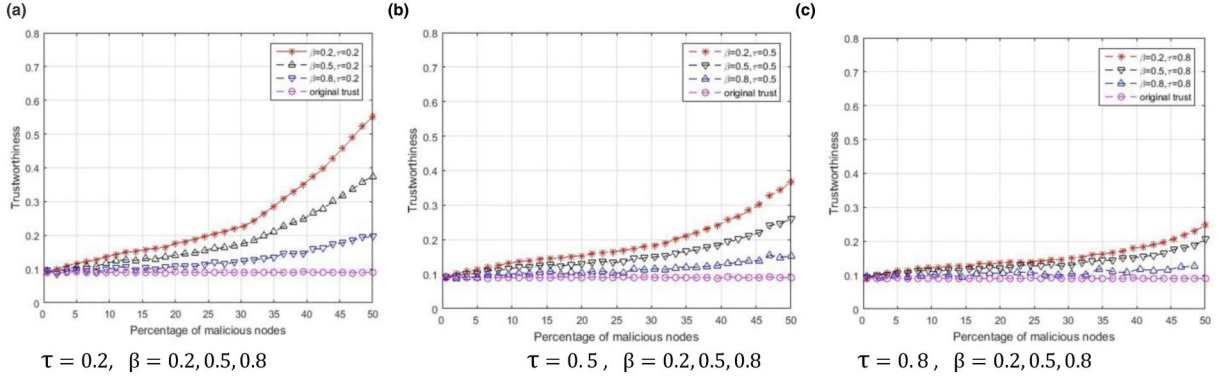


FIGURE 9 Addressing ballot-stuffing attack. (a) $\tau = 0.2, \beta = 0.2, 0.5, 0.8$, (b) $\tau = 0.5, \beta = 0.2, 0.5, 0.8$ and (c) $\tau = 0.8, \beta = 0.2, 0.5, 0.8$

The weight of each recommendation is given on the basis of the credibility of the recommender node. In addition, the recommending entities are required to report both the past and the current behaviour of the service provider. On each round, the number of malicious peers increases. The peers who are malicious remain malicious till the end. The good recommender nodes provide trustworthiness between $[0.83, 0.9]$ and malicious recommender nodes in the range $[0, 0.2]$.

Figure 8a,b and c show that as the percentage of malicious nodes increases, the trustworthiness of the node as a service provider decreases. The proposed scheme does not allow the dishonest nodes to highly degrade the trustworthiness of the service provider by using the credibility factor of the recommender nodes.

The results show that the scheme does not allow malicious users to highly impact the trustworthiness of the service provider in the presence of malicious users up to 40%. If there are more malicious peers in the neighbourhood and more weightage is given to the indirect trust (10), then it will impact the reputation score of the trustworthy service provider more. In Figure 8a, with $\beta = 0.2$, more weightage is given to indirect trust (10). It can be seen in Figure 8a, when $\tau = 0.2$ and the percentage of malicious nodes increases, the service provider's trustworthiness is affected more. In Figure 8a, the weighting

factor ($\tau = 0.2$) represents that while computing the credibility of recommender k , more weightage is given to recommender k rating behaviour similarity with the average ($uR_d(j, C_x, t_r)$) of all the recommendations received from neighbour recommenders for service provider j in the context C_x (7). As shown in Figure 8c, if there are more malicious peers in the neighbourhood, accuracy is improved if, while computing recommender credibility, more weight ($\tau = 0.8$) is given to rating behaviour similarity between the recommender and service consumer (Equation 7). It can be seen from Figure 8a,b and c, as τ decreases and the number of malicious users increases, the trustworthiness of the service provider is impacted more. Figure 8a illustrates that if more or equal weight is given to direct trust, the service provider trustworthiness is not impacted much.

6.7 | Experiment 7: Addressing ballot-stuffing attack

Figure 9 illustrates the effect of ballot-stuffing on the trustworthiness of a service provider with different percentages of malicious nodes.

This experiment is conducted over a network of 40 nodes. The experiment shows the results of a ballot-stuffing attack on

a single node having a peer's range of 40 nodes. The non-malicious recommender nodes provide the trustworthiness score between $[0, 0.2]$ and malicious recommender nodes in the range $[0.83, 1]$. In addition, the recommending entities are required to report both the past and current behaviour of the service provider.

In this set of experiments, when $\beta > 0.5$, more weightage is given to direct trust (in Equation 10). When $\tau < 0.5$, while computing the credibility of recommender k , more weightage is given to recommender k rating behaviour similarity with the average ($uR^{T_d(j, C_x, \tau_r)}$) of all the recommendations received from neighbour recommenders for service provider j in the context C_x (Equation 7). It can be seen in Figure 9 that as τ decreases and the number of malicious nodes increases, the trustworthiness of the service provider is impacted (raised) more. The results in Figure 9a show that the TRM protocol does not allow malicious users to highly raise the trustworthiness of the service provider in the presence of malicious users.

7 | CONCLUSION AND REMARKS ON THE FUTURE

With IoT's growing market, service consumers face problems associated with picking the most reliable service provider or service. Evaluating strangers' service providers' trustworthiness in IoT systems has become an important issue. Trust mechanisms stimulate cooperation among distributed entities. The TRM system is considered in IoT for providing qualified services and facilitating IoT entities, detecting malfunctions, and establishing proper collaborations. The risks of interactions are greatly reduced by TRM systems by helping consumers evaluate service provider quality before transactions. Because of the high impact of TRM systems, attacks that attempt to mislead service consumer decisions through dishonest recommendations are popular. Service providers may provide service with varying quality over time, periodically, or randomly.

This paper proposes a trust-based attack-resistant trust model for fog-based IoT. The proposed trust model is equipped with mechanisms to meet some requirements of the TRM system for the IoT environment, including survivability, adaptivity, and scalability. The potential benefits of the proposed TRM scheme for managing trust and reputation relationships are analysed through simulation results. The proposed scheme is robust against several attacks on the TRM system. We have considered the following five trust-related attacks that can disrupt the TRM system: self-promoting, bad-mouthing, ballot-stuffing, opportunistic service, and discriminatory.

The published studies provide no explicit support concerning the interoperability of different TRMs on fog-nodes. Many research problems remain unresolved in the design and development of TRM protocols for IoT systems—for example, privacy, context awareness, interoperability between TRMs, survivability, and robustness.

ORCID

Farhana Jabeen  <https://orcid.org/0000-0001-5420-4103>

REFERENCES

1. Nurelmadina, N., et al.: A systematic review on cognitive radio in low power wide area network for industrial IoT applications. *Sustainability*. 13, 338 (2021). <https://doi.org/10.3390/su13010338>
2. Jabeen, F., Nawaz, S.: In-network wireless sensor network query processors: state of the art, challenges and future directions. *Inf. Fusion*. 25, 1–15 (2015)
3. Jabeen, F., Nawaz, S.: In-network distributed event boundary computation in wireless sensor networks: challenges, state of the art and future directions. *KSII Trans. Internet Inf. Syst.* 7(11), 2804–2823 (2013)
4. Khanna, A., Kaur, S.: Internet of things (IoT), applications and challenges: a comprehensive review. *Wireless Pers. Commun.* 114, 1687–1762 (2020)
5. Jabeen, F., et al.: Trust and reputation management in healthcare systems: taxonomy, requirements and open issues. *IEEE Access*. 6, 17246–17263 (2018)
6. Panda, C.K., Bhatnagar, R.: Social internet of things in agriculture: an overview and future scope. In: *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*, 1st edn., pp. 317–334. Springer, London (2020)
7. Djedjig, N., et al.: Trust Management in the Internet of Things. In: Yassine, M., Abdellah, E., Mustapha, B. (eds.) *Security and Privacy in Smart Sensor Networks*, pp. 122–146. IGI Global, Hershey (2018). <https://doi.org/10.4018/978-1-5225-5736-4.ch007>
8. Sharma, A., et al.: Towards trustworthy Internet of Things: a survey on trust management applications and schemes. *Comput. Commun.* 160, 475–493 (2020)
9. Bao, F., Chen, I., Guo, J.: Scalable, adaptive and survivable trust management for community of interest-based Internet of Things systems. In: *Proceedings of the IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*, pp. 1–7. Mexico (2013)
10. Saied, Y.B., et al.: Trust management system design for the Internet of Things: a context-aware and multi-service approach. *Comput. Secur.* 39, 351–365 (2015)
11. Nitti, M., Girau, R., Atzori, L.: Trustworthiness management in the social internet of things. *IEEE Trans. Knowl. Data Eng.* 26(5), 1253–1266 (2014)
12. Grandison, T., Sloman, M.: A survey of trust in internet applications. *IEEE Commun. Surv. Tutorials*. 3, 2–16 (2000)
13. Memon, I., et al.: Protect mobile travelers information in sensitive region based on fuzzy logic in IoT technology. *Secur. Commun. Network*. 2020, 8897098 (2020). <https://doi.org/10.1155/2020/8897098>
14. Jayasinghe, U., Lee, H., Lee, G.M.: A computational model to evaluate honesty in social Internet of Things. In: *SAC '17 Proceedings of the Symposium on Applied Computing*, pp. 1830–1835. Marrakech (2017)
15. Liu, Y.C., Lianu, Y.H.: P2P dynamic trust model based on contextual factors. *J. Commun.* 37, 34–44 (2016)
16. Guo, J., Chen, I.R.: A classification of trust computation models for service-oriented internet of things systems. In: *Proceedings of the IEEE International Conference on Services Computing (SCC)*, pp. 324–331. San Francisco (2015)
17. Najib, W., Sulisty, S.W.: Survey on trust calculation methods in internet of things. *Procedia Comput. Sci.* 161, 1300–1307 (2019)
18. Xiao, H., Sidhu, N., Christianson, B.: Guarantor and reputation-based trust model for social internet of things. In: *Proceedings of the Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 600–605. Dubrovnik (2015)
19. Bao, F., Chen, I.R.: Dynamic trust management for internet of things applications. In: *Proceedings of the 2012 International Workshop on Self-aware Internet of Things (Self-IoT'12)*, pp. 1–6. ACM, San Jose (2012)
20. Chen, I.R., Guo, J., Bao, F.: Trust management for SOA-based IoT and its application to service composition. *IEEE Trans. Serv. Comput.* 9(3), 482–495 (2016)

21. Chen, I.R., Bao, F., Guo, J.: Trust-based service management for social internet of things systems. *IEEE Trans. Dependable Secur. Comput.* 13(6), 684–696 (2016)
22. Yasir, H., et al.: Context aware trust and reputation model for Fog based IOT. *IEEE Access.* 8, 31622–31632 (2010)
23. Shayesteh, B., Hakami, V., Akbari, A.: A trust management scheme for IoT-enabled environmental health/accessibility monitoring services. *Int. J. Inf. Secur.* 19, 93–110 (2020)
24. Yuan, J., Li, X.: A reliable and lightweight trust computing mechanism for IoT edge devices based on multi-source feedback information fusion. *IEEE Access.* 6, 23626–23638 (2018)
25. Shi, W.S., Zhang, X.Z., Wang, Y.F.: Edge computing: current situation and prospects. *Comput. Res. Dev.* 56, 73–93 (2019)
26. Junejo, H., et al.: A privacy-preserving attack-resistant trust model for internet of vehicles ad hoc networks. *Sci. Program.* 2020, 1058–9244 (2020). <https://doi.org/10.1155/2020/8831611>
27. Carolina, V., Mendoza, L., Kleinschmidt, J.H.: Mitigating on-off attacks in the internet of things using a distributed trust management scheme. *Int. J. Distributed Sens. Netw.* 11(11), 1–8 (2015)
28. Gu, L., Wang, J., Sun, B.: Trust management mechanism for internet of things. *China Commun.* 11(2), 148–156 (2014)
29. Truong, N.B., et al.: Toward a trust evaluation mechanism in the social internet of things. *Sensors.* 17(6), 1–24 (2017)
30. Truong, N., et al.: A reputation and knowledge based trust service platform for trustworthy social internet of things. In: *Proceedings of the Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pp. 930–937. Paris (2016)
31. Kowshalya, A.M., Valarmathi, M.L.: Trust management in the social internet of things, *Wireless Pers. Commun.* 96(2), 2681–2691 (2017)
32. Wenping, K., et al.: An efficient and credible multi-source trust fusion mechanism based on time decay for edge computing. *Electronics.* 9(3), 502 (2020)
33. Farhan, A., Awais, A., Sang, G.: Towards trust and friendliness approaches in the social internet of things. *Appl. Sci.* 9, 166–180 (2019)
34. Atzori, L., et al.: The social internet of things (SIOT) when social networks meet the internet of things: concept, architecture and network characterization. *Comput. Network.* 56(16), 3594–3608 (2012)
35. Rad, M.M., et al.: Social Internet of Things: vision, challenges, and trends. *Hum.-centric Comput. Inf. Sci.* 10(52) (2020)
36. Chen, Z., et al.: A scheme of access service recommendation for the social Internet of Things. *Int. J. Commun. Syst.* 29, 694–706 (2016)
37. Nitti, M., et al.: A subjective model for trustworthiness evaluation in the social Internet of Things. In: *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 18–23. Sydney (2012)
38. Tormo, G.D., Mármol, F.G., Perez, G.M.: Dynamic and flexible selection of a reputation mechanism for heterogeneous environments. *Future Generat. Comput. Syst.* 49, 113–124 (2015)
39. Abderrahim, O.B.: TMCoi-SIOT: A trust management system based on communities of interest for the social Internet of Things. In: *Proceedings of the 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 747–752. Spain (2017)
40. Abderrahim, O.B., Saidane, L.: CTMS-SIOT: A context-based trust management system for the social internet of things. In: *Proceedings of the 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1903–1908. Spain (2017)
41. Gai, F., et al.: Trust on the rate: a trust management system for social internet of vehicles. *Wireless Commun. Mobile Comput.* 2017, 11 (2017)
42. Jayasinghe, U., et al.: RpR: a trust computation model for social internet of things. In: *Proceedings of the International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld)*, pp. 930–937. Toulouse (2016)
43. Nelly, L., Avrachenkov, K.: The effect of new links on Google Pagerank. *Stoch. Model.* 22, 350–364 (2006)
44. Iqbal, R., et al.: Trust management in social Internet of vehicles: factors, challenges, blockchain, and Fog solutions. *Int. J. Distributed Sens. Netw.* 15(1), 1–19 (2019)
45. Um, T.W., et al.: Design and implementation of a trust information management platform for social internet of things environments. *Sensors.* 19(21), 4707 (2019)
46. Quantiles: http://www.statsdirect.com/help/nonparametric_methods/quantiles.htm. Accessed June 2020
47. Pham-Gia, T., Hung, T.L.: The mean and median absolute deviations. *Math. Comput. Model.* 34, 921–936 (2001)
48. Shafer, D.S., Zhang, Z.: *Descriptive Statistics. Beginning Statistics.* <http://2012books.lardbucket.org/books/beginning-statistics/s06-05-the-empirical-rule-and-chebyshev.html> (2012). Accessed April 2021
49. Riley, G.F., Henderson, T.R.: The ns-3 network simulator. In: Wehrle, K., Günes, M., Gross, J. (eds.) *Modeling and Tools for Network Simulation.* Springer, Berlin (2010)
50. Hawa, M.: A Graphical user interface for the ns-3 simulator. In: *Proceedings of the 12th International Conference on Computer Modeling and Simulation (ICCMS '20)*, pp. 159–163. Brisbane (2020)
51. Passarella, A., Conti, M.: Characterizing aggregate inter-contact times in heterogeneous opportunistic networks. In: *Proceedings of the 10th International IFIP Conference on Networking*, pp. 301–313. Spain (2011)
52. Karagiannis, T., LeBoudec, J., Vojnovi, M.: Power law and exponential decay of intercontact times between mobile devices. *IEEE Trans. Mobile Comput.* 9(10), 1377–1390 (2010)

How to cite this article: Jabeen F, Khan ZR, Hamid Z, Rehman Z, Khan A. Adaptive and survivable trust management for Internet of Things systems. *IET Inf. Secur.* 2021;1–20. <https://doi.org/10.1049/ise2.12029>